Int J Data Sci Anal (2017) 4:251–264
DOI 10.1007/s41060-017-0052-3

REGULAR PAPER

The Matsu Wheel: a reanalysis framework for Earth satellite imagery in data commons

Maria T. Patterson[1] · Nikolas Anderson[1] · Collin Bennett[2] · Jacob Bruggemann[1] · Robert L. Grossman[1] · Matthew Handy[3] · Vuong Ly[3] · Daniel J. Mandl[3] · Shane Pederson[2] · James Pivarski[2] · Ray Powell[1] · Jonathan Spring[1] · Walt Wells[4] · John Xia[1]

Received: 21 June 2016 / Accepted: 19 March 2017 / Published online: 30 March 2017

Abstract Project Matsu is a collaboration between the Open Commons Consortium and NASA focused on developing open source technology for the cloud-based processing of Earth satellite imagery and for detecting fires and floods to help support natural disaster detection and relief. We describe a framework for efficient analysis and reanalysis of large amounts of data called the Matsu "Wheel" and the analytics used to process hyperspectral data produced daily by NASA's Earth Observing-1 (EO-1) satellite. The wheel is designed to be able to support scanning queries using cloud computing applications, such as Hadoop and Accumulo. A scanning query processes all, or most, of the data in a database or data repository. In contrast, standard queries typically process a relatively small percentage of the data. The wheel is a framework in which multiple scanning queries are grouped together and processed in turn, over chunks of data from the database or repository. Over time, the framework brings all data to each group of scanning queries. With this approach, contention and the overall time to process all scanning queries can be reduced. We describe our Wheel analytics, including an anomaly detector for rare spectral signatures or anomalies in hyperspectral data and a land cover classifier that can be used for water and flood detection. The resultant products of the analytics are made accessible through an API for further distribution. The Matsu Wheel allows many shared data services to be performed together to efficiently use resources for processing hyperspectral satellite image data and other, e.g., large environmental datasets that may be analyzed for many purposes.

Keywords Earth satellite data · Reanalysis framework · Data commons

1 Introduction

Although large amounts of satellite imagery data are available for download from NASA, as well as from other sources, by and large, these large volumes of data are not easy to process and analyze for many researchers. Scientists are finding that the bottleneck to discovery is no longer a lack of data but an inability to manage and analyze large datasets. Imaging from Earth satellites in particular can quickly produce large volumes of data, necessitating a computing environment capable of scientific analysis of many images.

Project Matsu is a collaborative effort between the Open Science Data Cloud (OSDC), managed by the Open Commons Consortium (OCC), and NASA, working to alleviate these difficulties by developing open source tools for processing and analyzing Earth satellite imagery in cloud-based "data commons," or cyber-infrastructure that colocate data with storage and commonly used data access and analysis tools [2]. The goals of Project Matsu are to: (1) Develop an open source cloud-based infrastructure to process Earth satellite image data with API-accessible services, (2) Develop parallel algorithms and analytics using Hadoop's MapReduce and related frameworks for processing large amounts

This paper is an extension version of the BDS2016 accepted paper "The Matsu Wheel: A Cloud-based Framework for the Efficient Analysis and Reanalysis of Earth Satellite Imagery" [1].

✉ Maria T. Patterson
mtpatter@uchicago.edu

[1] Center for Data Intensive Science, University of Chicago, Chicago, IL 60637, USA

[2] Open Data Group, River Forest, IL 60305, USA

[3] NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA

[4] Open Commons Consortium, Chicago, IL, USA

of satellite image data to detect floods and other events for disaster assistance and relief, and (3) Operate persistent cloud-based services that process satellite image data and other data sources each day and make the resulting reports available to the research community and other interested parties.

The Matsu "Wheel" is a framework we have developed for simplifying Earth satellite image analysis on large volumes of data by providing an efficient system that performs all the common data services and then passes the prepared chunks of data in a common format to the analytics, which process each new chunk of data in turn.

This paper is organized as follows: In the subsections below, we provide motivation for an analytic wheel framework and introduce the hyperspectral satellite data to which we apply this framework. In Sect. 2 we put this analytic wheel approach and the algorithms applied in context of related work. In Section 3, we describe the workflow of our analytic wheel implementation in our data commons, providing details on the computational environment in which it runs and the preprocessing applied to the data in 3.1 and 3.2. The architecture of the wheel, the flow of data through the wheel analytics, and design considerations are discussed in Sects. 3.3 and 3.4. In Sect. 4, we describe the structure of an individual analytic in the wheel and detail each of the algorithms the Matsu Wheel runs, including several anomaly detection algorithms and a classification algorithm. We discuss efficiencies in our application and directions this work can be extended in Sect. 5 and close with a summary in Sect. 6.

1.1 Motivation for an analytic wheel

A common class of problems require applying an analytic computation over an entire dataset. Sometimes these are called *scanning queries* since they involve a scan of the entire dataset. For example, analyzing each image in a large collection of images is an example of a scanning query. In contrast, standard queries typically process a relatively small percentage of the data in a database or data repository.

With multiple scanning queries that are run within a time that is comparable to the length of time required for a single scan, it can be much more efficient to scan the entire dataset once and apply each analytic, in turn, versus scanning the entire dataset for each scanning query as the query arrives. This is the case unless the data management infrastructure has specific technology for recognizing and processing scanning queries. In this paper, we introduce a software application called the Matsu Wheel that is designed to support multiple scanning queries over satellite imagery data.

The motivation for using an analytic wheel is that by performing all of the data processing services together on chunks of data, a data commons can more efficiently use resources, including available network bandwidth, access to secondary storage, and available computing resources. The wheel consolidates operations that are shared by multiple analytic pipelines, such as data loading, and applies them together before scanning queries process the data. This is especially important with reanalysis of data in a repository, in which the entire dataset is processed using an updated algorithm, a recalibration of the data, a new normalization of the data, a new workflow, etc. The motivation behind the analytic wheel is to streamline and share these services so that they are only performed once on each chunk of data in a scanning query, regardless of the number or type of scanning analytics run over the data, so that new scanning analytics can be added at minimal cost.

The wheel analytic approach differs from common workflows around data accessed from repositories. Traditionally, individual researchers may use stand-alone workstations to download datasets and run an analytic in isolation. This may be possible for smaller datasets but can be difficult to impossible for large datasets exceeding the resources of individual workstations.

For data stored in a relational database, a potentially long-running query is needed to fetch the dataset for each analytic, which is especially inefficient for scanning queries which access all or large subsets of datasets. Multiple scanning queries in this case adds the additional complexity inherent in concurrent retrievals of the same data.

In a compute-on-demand cloud, virtual machines may be launched to run specific analytics over data that are mounted to the instance. In this case, again, contention issues may arise as multiple virtual machines concurrently access the same data mount. Additionally, for data that have preprocessing requirements that may be the same for several commonly applied analytics, it is inefficient for each analytic to separately apply preprocessing.

While each analysis framework has its distinct advantages, the Matsu Wheel system provides a significantly more efficient use of resources over alternative methods (1) for datasets that are large and may be continuously growing or updated, (2) for datasets that may require preprocessing (e.g., commonly applied corrections or normalizations), and (3) for datasets where multiple analytics are applied to the same data and the number of scanning queries grows. These characteristics are generally the case for Earth satellite imagery data.

1.2 Application to Earth satellite data

In Earth satellite and hyperspectral image data, especially, the Matsu Wheel system can increase the efficiency of a data analytics system. Earth satellite data can be large and have high-volume throughput as new data are continuously acquired for active satellite missions. Earth satellite data also may require heavy preprocessing common to many types of applications, such as atmospheric corrections or geographic

coordinate corrections. In these types of use cases, using a system like the Matsu Wheel system can consolidate operations by grouping them for multiple analytics.

The Matsu Wheel is currently used to process the data produced each day by NASA's Earth Observing-1 (EO-1) satellite and makes a variety of data products available to the public. In addition to the Atmospheric Corrector, the EO-1 satellite has two primary scientific instruments for land observations, the Advanced Land Imager (ALI) and a hyperspectral imager called Hyperion [3,4]. EO-1 was launched in November 2000 as part of NASA's New Millennium Program (NMP) initiative for advancing new technologies in space and is currently in an extended mission.

The ALI instrument acquires data in 9 different bands from 0.48–2.35 μm with 30-meter resolution plus a panchromatic band with higher 10-meter spatial resolution. The standard scene size projected on the Earth surface equates to 37 km × 42 km (width × length). The Hyperion instrument has similar spatial resolution but higher spectral resolution, observing in 242 band channels from 0.357–2.576 μm with 10-nm bandwidth. Hyperion scenes have a smaller standard footprint width of 7.7 km.

The Matsu Wheel runs analytics over Level 1G data, which are data that have several common corrections applied and are also transformed to Geographic Tagged Image File Format (GeoTiff) format. These data are radiometrically corrected, resampled for geometric correction, and registered to a geographic map projection. The GeoTiff data and metadata for all bands in a single Hyperion scene can amount to 1.5–2.5 GB of data for only the Level 1G data. The total size of ALI and Hyperion Level 1G GeoTiff data over one year (2015) is about 11 terabytes, and a similar amount of data is produced by each analytic that creates GeoTiff data products. A cloud environment for shared storage and computing capabilities is ideal for scientific analysis of many scenes, which can quickly grow to a large amount of data.

2 Related work

The Matsu Wheel approach differs from other common data management or data processing systems. Real-time distributed data processing frameworks (for example, Storm, S4, or Akka) are designed to process data in real time as it flows through a distributed system (see also, e.g., [5–7]). In contrast, the wheel is designed for the reanalysis of an entire static dataset that is stored in distributed storage system (for example, the Hadoop Distributed File System) or distributed database (for example, HBase or Accumulo).

A number of data flow systems exist for simplifying large-scale data processing, and particularly over Hadoop storage systems. For example, the Apache Pig project provides a system for applying SQL-like queries to build complex data flows over Hadoop and is designed for performing long series of data operations [8]. In contrast, the wheel approach is for independent operations that have a need to be applied to all data and whose results should be updated with the addition of new data. Oozie also provides a way to bundle Hadoop jobs into work flows [9]. In contrast, the wheel approach is designed for preprocessing of common heavy tasks and allowing individual analytics to be written without having to be concerned with interacting with other tasks. Additionally, Spark is a tool that adds in-memory data processing to Hadoop enabling multiple applications to share datasets [10], whereas the wheel approach is to remove common operations from analytics by preprocessing data and the pass data to the analytics in tandem.

It is important to note that any distributed scale-out data processing system based upon virtual machines has certain performance issues due to the shared workload across multiple virtual machines associated with a single physical node. In particular, these types of applications may have significant variability in performance for real scientific workloads [11]. This is true, when multiple scanning queries hit a distributed file system, a NoSQL database, or a wheel-based system on top of one these infrastructures. When a NoSQL database is used for multiple scanning queries with a framework like the wheel, the NoSQL database can quickly become overloaded.

For the context of the individual Matsu Wheel analytics presented here, Griffin et. al. [12] provide an overview of example analyses of data from the EO-1 satellite and of hyperspectral imaging data in general. Analysis of hyperspectral data falls primarily into one of two categories. The first class is terrain characterization, which can be approached more generally as a classification problem applied to the spectrum at each pixel. Related work in remote sensing has been done to develop algorithms for cloud detection and land coverage classification, with applications to agriculture, forestry, mineral mapping, vegetation studies, and biodiversity (see, e.g., [13–15]). The second class is source/object detection, which is mainly applied to detect rare objects against a heterogeneous background and can subsequently be framed as an anomaly detection problem (see, e.g., [16] and references therein). Given the variety of analyses that can be applied to the same hyperspectral imaging data, these use cases are particularly well suited to an analytic wheel framework, and we present both anomaly detection and land coverage classification types of analyses in Sect. 4.

3 Analytic wheel workflow

3.1 Computational environment

The computing infrastructure for the daily processing of EO-1 data for Project Matsu involves both an OpenStack-based

computing platform for preprocessing and a Hadoop-based computing platform for the Wheel analytic scanning queries, both of which are managed by the OCC (www.occ-data.org) in conjunction with the University of Chicago. The OpenStack platform (the Open Science Data Cloud [17]) currently contains 60 nodes, 1208 compute cores, 4832 GB of compute RAM, and 1096 TB of raw storage. The Hadoop [18] platform currently contains 28 nodes, 896 compute cores, 261 TB of storage, and 3584 GB of compute RAM.

3.2 Preprocessing of data on the OSDC

The OpenStack-based Open Science Data Cloud provides a number of cloud-based services for Project Matsu. As new EO-1 observations are received from NASA each day, the data are stored on a distributed, fault-tolerant file system (GlusterFS and Ceph) and preprocessed prior to the application of the wheel-based analytics. The images are converted into SequenceFile format, a file format more suited for MapReduce, and uploaded into the Hadoop Distributed File System (HDFS) [19]. Metadata and compute summary statistics are extracted for each scene and are stored in Accumulo, a distributed NoSQL database [20]. The metadata are used to display the geospatial location of scenes via a web map service so that users can easily visualize which areas of the Earth are observed in the data processed by the Matsu Wheel.

The Matsu data flow for processing EO-1 images and producing data and analytic products is described below:

1. Performed by NASA/GSFC as part of their daily operations:
   (a) Transmit data from NASA's EO-1 Satellite to NASA ground stations and then to NASA/GSFC.
   (b) Align data and generate Level 0 images.
   (c) Transmit Level 0 data from NASA/GSFC to the OSDC.
2. Run by NASA on the OSDC OpenStack cloud for Matsu and other projects:
   (a) Store Level 0 images in the OSDC Public Data Commons for long-term, active storage.
   (b) Within the OSDC, launch Virtual Machines (VMs) specifically built to render Level 1 images from Level 0. Each Level 1 band is saved as a distinct image file (GeoTiff).
   (c) Store Level 1 band images in the OSDC Public Data Commons for long-term storage.
3. Run specifically for Project Matsu and the Wheel on the Hadoop cloud:
   (a) Read Level 1 images, combine bands, and serialize image bands into a single file.
   (b) Store serialized files on HDFS.
   (c) Run wheel analytics on the serialized Level 1 images stored in HDFS.
   (d) Store the results of the analysis in Accumulo for further analysis, generate reports, and load analytic results into a Web Map Service for programmatically accessible distribution.

3.3 Analytic "wheel" architecture

The analytic wheel is so named because multiple analytics are applied to all of the data one chunk of data at a time, roughly analogous to the way a Ferris Wheel works. By the time a Ferris Wheel makes a complete revolution, all of the chairs have been processed exactly one time. With big data, retrieving or processing data multiple times is an inefficient use of resources and should be avoided.

The Matsu Wheel system is unique in that, essentially, the *data* are flowing through the framework while the analytic queries sit in place and wait for new data to scan. This type of scanning framework, in which the data only need to be read once for multiple analytics, is increasingly important as scientific datasets become larger, new data are acquired at a faster rate, and data are used and reused for many purposes.

3.3.1 Key components of the data commons environment

In Sect. 1.1, we discussed how an analytic wheel may be more efficient for scanning query type analytics that process all or most of the data in a dataset. Specifically, an analytic wheel framework is well suited for the processing and analysis or reanalysis of data in a data commons, which is cyber-infrastructure that colocates data archives with storage and computing resources and commonly used tools for analyzing and sharing data to create a resource for the research community [2].

The architecture of a data commons requires consideration of balancing the needs of both data archiving for persistent storage and also computation on an active system where data may be physically moved, updated, or distributed across different resources. To manage references and queries to pieces of data to be processed by the analytic wheel, we assign digital identifiers to the location (url) of each scene (image file) to be analyzed. Data are then accessed via digital identifiers through a simple tracking and index server with a REST-like API interface.

These identifiers are returned through the available images map at matsu.opensciencedatacloud.org that allows for simple search for data of interest by date of observation, location, and overall radiance (brightness) level. Each scanning analytic in the wheel refers to batches of scenes accessed by their digital identifiers and subsequently uses

the identifiers as the keys for entries in each analytic's summary results database table, as described in the next section.

3.3.2 Running the Matsu Wheel analytics

The EO-1 Satellite collects data daily, and lower level processing is performed over the raw data in the commons, as described in Sect. 3.2, to generate the Level 1 data used by the analytic wheel.

To detect and stage new Level 1 data to be processed by the wheel, we currently use Apache Storm, a distributed and fault-tolerant computation framework for stream processing. Storm topologies continuously watch for newly available Level 1 GeoTiff files, convert these data to SequenceFile format and load into HDFS, index the data by applying digital identifier to scenes, and update an Accumulo database tracking all images available in the commons for processing.

When a new batch of data becomes available in HDFS as part of the preprocessing described above, the MapReduce scanning analytics in the wheel kick off. In our implementation, the Matsu Wheel is run daily after receiving the previous day's observations. To run the wheel over a daily batch, the digital identifiers for that day's observations are passed to the start of the wheel, and each analytic in the wheel in turn processes the batch. Intermediate output is written to HDFS, and each analytic's results are stored in an Accumulo table as JSON. Secondary analysis that may process the results of other analytics can be done "off the wheel" by using the Accumulo-stored JSON as input.

As many analytics can be included in the wheel as can run in the allowed time. If new data are obtained each day, then the limit is 24 h to avoid back-ups in processing. For other use cases, there may be a different time window in which the results are needed. This can be seconds, minutes, or hours. Our MapReduce analytic environment is not designed to yield immediate results, but the analytics can be run on individual images at a per minute speed. Analytics with results that need to be made available as soon as possible can be configured to run first in the wheel. Similarly, downstream analytics can use the results of other analytics by appropriate placement running in the wheel. We show a diagram of the flow of EO-1 satellite data from acquisition and ingest into HDFS through the analytic wheel framework in Fig. 1.

The wheel architecture is an efficient framework not restricted only to image processing or to daily batch processing, but is applicable to any workflow where an assortment of analytics is applied to data that require heavy preprocessing, have high-volume throughput, or may be regularly reprocessed or reanalyzed.

3.4 Design considerations of the "wheel"

The analytic wheel is designed so that different analytics are batched together, and each geospatial region is accessed once by the wheel and the batch of analytics is applied. In the current design, virtual machines are used for the processing of each geospatial region. If there is a desire to complete a turn of the wheel in a fixed period of time (for example once a day), then as the complexity of the analytics increases, the data can be chunked at a finer resolution and additional virtual machines can be used to process the data. As a simple example, change detection can be implemented in this way by simply using a virtual machine to process all the data for a given spatial region over a period of time. Once the basic satellite images are processed, finer and finer spatial resolutions can be used to increase the scale of parallelism to process all of the data in a fixed period of time for all of the analytics being applied as part of the change detection algorithm. If necessary, the set of analytics being applied to a given spatial region being fetched by the wheel can be split into two or more sub-batches, with each sub-batch processed by a separate virtual machine in parallel. This type of application was one of the motivations for the wheel architecture.

In general when accessing data from databases, it is usually significantly faster to access the data using an indexed scan. On the other hand, when it is required to evaluate each record in a database (a "full table scan"), then using the optimizations provided by a full table scan versus an indexed scan is significantly faster [21]. Since a disk can only access one contiguous sequence of blocks at a time, as the number of concurrent full table scans increases, it becomes more and more advantageous to batch the computations together and apply them to the data once it is loaded into memory versus separately fetching the blocks of data into memory for each analytic separately. The Matsu Wheel essentially uses the same optimizations but applies them currently to each virtual machine accessing distinct geospatial regions.

4 Analytics

In this section, we describe the Matsu Wheel analytics currently in place. We are running five scanning queries on daily images from NASA's EO-1 satellite with the Matsu Wheel, including several spectral anomaly detection algorithms and a land cover classification analytic.

4.1 Analytic requirements

Generally, an analytic wheel framework requires that each scanning query analytic take as input a batch of data. Existing queries then plug in easily and may need only small modifications before being included in the wheel. Queries do not

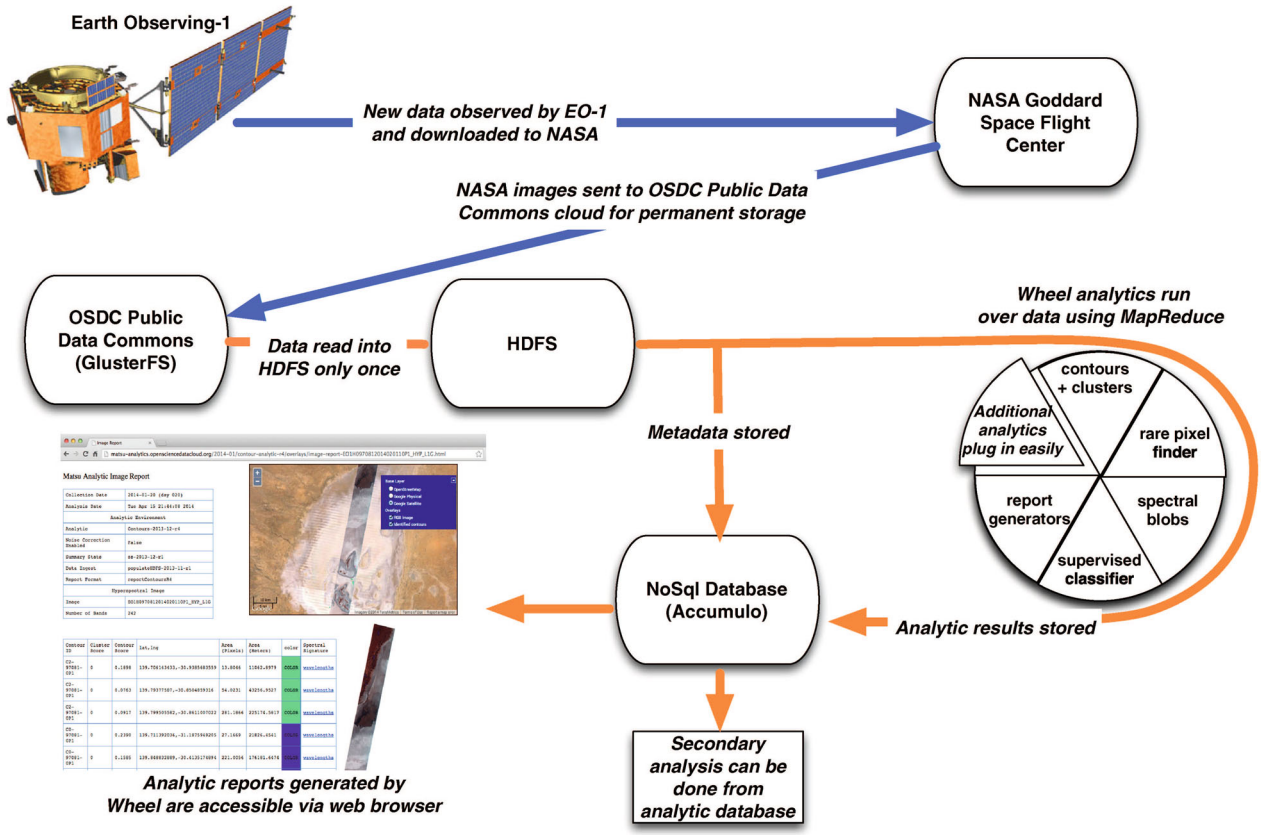


Fig. 1 A diagram of the flow of EO-1 ALI and Hyperion data from data acquisition and ingest through the Matsu Wheel framework. *Orange* denotes the processes performed on the OSDC Hadoop cloud. With the wheel architecture, the data need to be read in only once regardless of the number of analytics applied to the data. The Matsu Wheel system is unique in that, essentially, the *data* are flowing through the framework while the analytic queries sit in place and scan for new data. Additional analytics plug in easily, with the requirement that an analytic takes as input a batch of data to be processed. An analytic query may be written such that it can be run on its own in the wheel or written to take as input the output of an upstream analytic. The report generators are an example of the latter case, generating summary information from upstream analytics

need to be ordered in the wheel unless the query references results from a query that must be run earlier or unless the query is slotted early in the wheel for time considerations.

Our implementation of the analytic wheel for Project Matsu reads and processes data with Hadoop and stored results in JSON format written to Accumulo tables. For additional analytics to be included in the Matsu Wheel, a query must take as input a list of sequential files in .seqpng format stored in HDFS. The query then executes the analytic(s) and writes output results (e.g., cluster size, cluster location in latitude and longitude, anomaly score) in JSON format to an Accumulo table using the assigned identifier for the analyzed scene as the database entry key.

4.2 Contours and Clusters

The Contours and Clusters analytic produces contours of geospatial regions, showing clusters of pixels with rare spectral signatures. The contours are false-color outlines of regions, and the darker the color, the higher the density of pixels with the specific signature.

The input data consist of Level 1G EO-1 scenes from the Hyperion instrument, essentially a set of radiances for all spectral bands for each pixel in an image. The radiance in each band is then divided by its underlying solar irradiance to convert to units of reflectivity or at-sensor reflectance. This is done by scaling each band individually by the irradiance and then applying a geometric correction for the solar elevation and Earth–Sun distance, as shown in Eq. 1,

$$\rho_i = \left(\frac{\pi}{\mu_0 F_{0,i} / d^2_{\text{Earth}-\text{Sun}}} \right) L_i \tag{1}$$

where ρ_i is the at-sensor reflectance at channel i, $\mu_0 = \cos(\text{solar zenith angle})$, $F_{0,i}$ is the incident solar flux at channel i, $d_{\text{Earth}-\text{Sun}}$ is the Earth–Sun distance, and L_i is the irradiance recorded at channel i [12]. This correction accounts for differences in the data due to time of day or year.

We then apply a principal component analysis (PCA) to the set of reflectivities and extract the top N (we choose $N = 5$) PCA components for further analysis. Pixels are clustered in the transformed 5-dimensional spectral space using a k-means clustering algorithm. For an individual image, $k = 50$ spectral clusters are formed and then ranked from most to least extreme using a normalized Mahalanobis distance metric to identify the most anomalous spectra. For each spectral cluster, adjacent pixels in an image are grouped together into contiguous objects based on the purity or fraction of pixels that belong to that cluster and are then ranked again based on their distance from the spectral cluster center.

For each cluster, two scores are produced, indicating (1) how anomalous the spectral signature is in comparison with the rest of the image and (2) how close the pixels within the contour are to the cluster signature. The top ten most anomalous clusters across all scenes in an analytic wheel batch are singled out for manual review and highlighted in a daily overview summary report.

The analytic returns the clusters as contours of geographic regions of spectral "anomalies" which can then be viewed as polygonal overlays on a map. The Matsu Wheel produces image reports for each image, which contain an interactive map with options for an OpenStreetMap, Google Physical, or Google Satellite base layer and an RGB image created from the hyperspectral data and identified polygon contours as options for overlays. The results can be easily accessed via daily image reports through a web browser.

This wheel analytic has successfully identified and rank ordered regions of interesting activity on the Earth's surface, including several volcanic events. We show an example analytic image report for one significant event detection in Fig. 2. This detection is from an Earth Observing-1 visit from late June 2016 of the Kilauea volcano in Hawaii, a few days prior to a flow eruption from a vent from the Pu'u 'O'o crater. The top left table shows summary information about the observation. The top right shows the interactive map overlay with a purple contour in the center, which encircles a large spectral anomaly. The cluster and contour scores for this object are shown in the table in the bottom left, and the spectrum of the area enclosed by the contour is shown in the bottom right.

4.3 Rare Pixel Finder

The Rare Pixel Finder (RPF) is another anomaly detector. This analytic is designed to find small clusters of unusual pixels in a hyperspectral image. The algorithm, as we implement it, is applied directly to the EO-1 data in radiances, but the data can also be transformed to reflectances or other metrics or can have logs applied.

Using the subset of k hyperspectral bands that are determined to be most informative, it computes k-dimensional Mahalanobis distances and finds the pixels most distant. From this subset, pixels that are both spectrally similar and geographically proximate are retained. Spectrally similar pixels that can be further grouped into small compact sets are reported as potential outlying clusters. Details of the different steps of the algorithm are given below.

In the preprocessing step, we remove areas of the image that are obvious anomalies not related to the image (e.g., zero radiances on the edges of images), as well as spectral bands that correspond to water absorption or other phenomena that result in near-zero observed radiances. Any transformations are applied to the data at this point, such as transforming radiance to reflectance or logs.

Once the data are preprocessed, the Mahalanobis distance (D_i) is calculated for each pixel. Then, only the subset S_1 of pixels that satisfy $D_i > k_1$ are selected, where k_1 is chosen such that S_1 only contains 0.1–0.5% of pixels. In practice, k_1 was based on the upper 6σ of the distribution of sample distances, assuming a log-normal distribution for the distances.

For the subset of S_1 pixels chosen in the previous step, we next compute a similarity matrix T with elements T_{ij} measuring the spectral similarity between each pair of pixels. The similarity metric is based on the dot product between each pair of points and measures the multi-dimensional angle between points. The matrix is only formed for the subset S_1 and contains a few hundred rows. The pixels are then further subset. Pixel i is selected for set S_2 if $T_{ij} > k_2$ for $j \neq i$. The parameter k_2 is chosen to be very close to 1; in this way, we select objects that are both spectrally extreme but still have similar spectral profiles to one another.

In order to cluster the pixels geographically, we assume they lie on a rectangular grid. An L_1 norm metric is applied to the subset S_2 in order to further whittle down the candidate set to pixels that are spectrally extreme, spectrally similar, and geographically proximate. The metric is set so that traversing from one pixel to an immediately adjacent one would give a distance of 1, as if only vertical and horizontal movements were allowed. Pixel i is selected for set S_3 if $M_{ij} < k_3$ for $j \neq i$ and for some small value of k_3. We used $k_3 = 3$, so that pixels either needed to be touching on a face or diagonally adjacent.

In the final step, a simple heuristic designed to find the connected components of an undirected graph is applied. We further restrict to a set S_4 that are geographically proximate to at least k_4 other pixels (including itself). We used $k_4 = 5$, with the goal of finding compact clusters of between 5 and 20 pixels in size. The clumping heuristic then returns the pixels that were mapped into a cluster, the corresponding cluster ID's, and the associated distances for the elements of the cluster.

The flagged pixels are reported as "objects." These groups of objects are further filtered in order to make sure that they actually represent regions of interest. The following criteria must be satisfied in order to qualify:

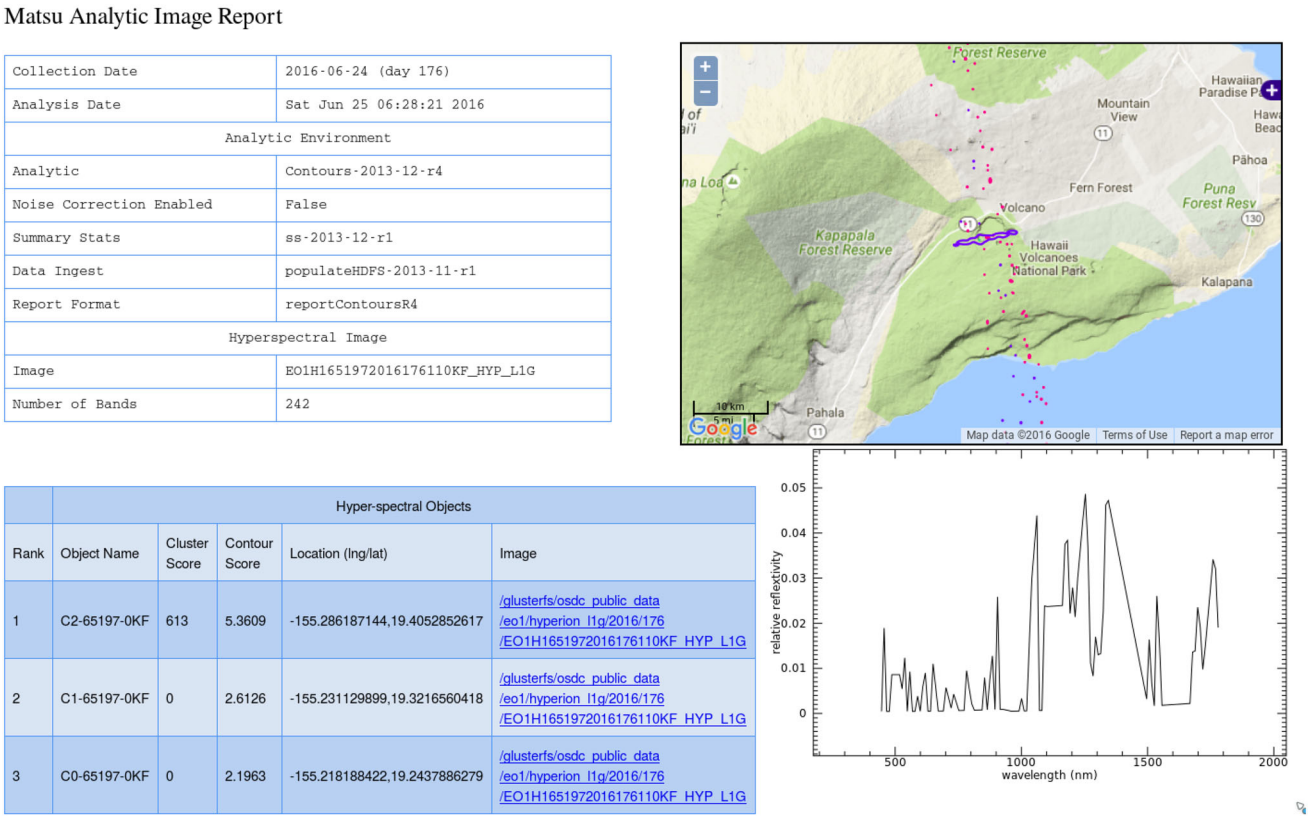


Fig. 2 A screenshot of a Matsu analytic image report for a Contours and Clusters spectral anomaly analytic that automatically identified regions of interest around the Kilauea volcano, confirmed as active by other sources in June 2016. The reports contain basic information about the scene analyzed and the analytic products and an interactive map with a color image made from the scene (not displayed) and analytic results available as overlays on an OpenStreetMap, Google Physical, or Google Satellite base layer. In this analytic, interesting regions are given a cluster score from 0 to 1000 based on how anomalous they are compared to the average detection and appear as colored contours over the image. The *bottom right* shows the spectrum for the anomalous region

1. *Spectral extremeness* The mean Mahalanobis distance must be greater than or equal to some parameter p_1. This selects clusters that are sufficiently extreme in spectral space.
2. *Spectral closeness* The signal-to-noise ratio must be greater than or equal to some parameter p_2. This selects clusters that have similar values of Mahalanobis distance.
3. *Cluster size* All clusters must be between a minimum value parameter p_3 and a maximum parameter p_4 pixels in size (the goal of this classifier is to find small clusters).
4. *Cluster dimension* All clusters must have at least some parameter p_5 rows and columns (but are not restricted to be rectangles).

Parameters p_1–p_5 are all tuning parameters that can be set in the algorithm to achieve the desired results. Examples of the parameters used can be shown in Table 1.

Table 1 Example parameter values for filtering steps in the Rare Pixel Finder analytic

Parameter	Value	Definition
p_1	2000	Spectral extremeness
p_2	5	Spectral closeness
p_3	4	Minimum cluster size
p_4	20	Maximum cluster size
p_5	2	Cluster dimension

4.4 Gaussian Mixture Model and K-Nearest Neighbors (GMM–KNN) algorithm

We apply the GMM–KNN analytic to data from the ALI instrument. The GMM–KNN algorithm is another analytic designed to find small clusters of unusual pixels in a multi-

spectral image. In short, using fewer spectral bands, it fits the most common spectral shapes to a Gaussian Mixture Model (smoothly varying, but makes strong assumptions about tails of the distribution) and also a K-Nearest Neighbor model (more detailed description of tails, but granular), then searches for pixels that are far from common. Once a set of candidate pixels have been found, they are expanded and merged into "clumps" using a flood-fill algorithm. The clumps are then characterized by deriving a suite of features via KNN, edge detection, and the distribution of pixel spectra in the clump. The clumps with the most unusual features are scored and rank ordered.

The first step in the GMM–KNN analytic is preprocessing specific to this analytic. The preprocessing steps are motivated by two assumptions about the input data: (1) Absorption or selective reflection of light is multiplicative and (2) the predominant source of variation in most images is intensity, not color. However, variations in intensity are transient, depending on changes in observation orientation, while variations in color are more indicative of real objects on the ground. To address this, we take the logarithm of the radiance value of each band and project the result onto a color-only basis to remove variations in intensity and highlight instead ground objects.

The next step is to fit a Gaussian Mixture Model of $k = 20$ Gaussian components to the spectra of all pixels. This is sufficiently large to cover the major structures in a typical image. A Gaussian Mixture Model generalizes clustering by allowing components to be asymmetrically shaped and allowing cluster inclusion to be nonexclusive. Since the Gaussian Mixture Model describes the major features of the distribution, pixels with spectra that are far from the Gaussian centroids are outliers. We use the log-likelihood of the GMM distribution as a "GMM outlier score."

The GMM score is derived purely from spectral shape, so we use a flood-fill to expand GMM outliers to enclose any surrounding region that is also anomalous and merge GMM outliers if they are in the same clump. We then characterize the merged multi-pixel multispectral clumps using a variant of K-Nearest Neighbors, in which we specify a radius in log-spectral space and count the number of spectra within that radius. In particular, we used a Gaussian measure with standard deviation r for $r = 0.2$ and $r = 0.5$. The count is not an integer because of the Gaussian measure, and it is normalized to the total number of non-mask pixels in the image to derive a "KNN outlier score." More anomalous spectra have lower scores (less density).

The spectral clumps detected in a given analytic wheel batch of images are rank ordered by their GMM and KNN outlier scores and presented in a summary report with spectral details available for individual outliers. We show an example summary report produced by the Matsu Wheel for the GMM–KNN analytic in Fig. 3.

4.5 Spectral Blobs

The Spectral Blobs analytic is used to identify anomalous "blobs" of regions that are similar within the region in spectral space but far from other blobs. Here, a "blob" is a set of pixels that are spectrally similar to one another and need not be spatially contiguous, unlike the Contours and Clusters analytics, which detects both spectral similarity and spatial contiguity.

The Spectral Blobs algorithm uses a "windowing" technique to create a boundary mask from the standard deviation of neighboring pixels. A catalog of blobs that contain varying numbers of pixels is created. Blobs are scored by comparing similarity to other blobs within the same image.

First, the analytic creates a boundary mask using a rolling window of 3-by-3 pixel windows. It uses an undirected graph search algorithm to separately label each group of spatially connected components of the mask. It then applies statistical significance tests (t-test, Chi-squared) to the spectral features of the blobs to test whether regions can be merged together, according to a tunable threshold. Spectrally similar regions are merged together.

The anomalous regions are the small spatial blobs that are not a member of any larger spatial cluster. A spectral dissimilarity score for each blob is generated by comparison to other blobs. A high score indicates increasing dissimilarity from other blobs in an image.

4.6 Supervised Spectral Classifier

The Supervised Spectral Classifier is a land coverage classification algorithm for the Matsu Wheel. We are particularly interested in developing these analytics for the detection of water and constructing flood maps to complement the onboard EO-1 flood detection system [22]. This analytic is applied to both ALI and Hyperion Level 1G data and classifies each pixel in an image as a member of a given class in a provided training set. We currently implement this analytic with a simple land coverage classification training set with four possible classes: clouds, water, desert / dry land, and vegetation.

The classifier takes as input the reflectance values of each pixel as the characterizing vector and applies a support vector machine (SVM) algorithm. In this implementation of the classifier, we bin Hyperion data to resemble ALI spectra for ease of use and computation speed in training the classifier. We construct a vector space from all ALI bands and two additional ALI band ratios, the ratios between ALI bands 3:7 and 4:8.

We make use of the SVM method provided by the Python scikit-learn machine learning package [23], which provides multi-class classification support via a "one-against-one" approach in which multiple classifiers are constructed, each

Matsu Analytic Summary Report for 2016-09-27 (Julian day 2016 271)

Analysis date: Wed Sep 28 08:55:51 2016

Analytic Environment

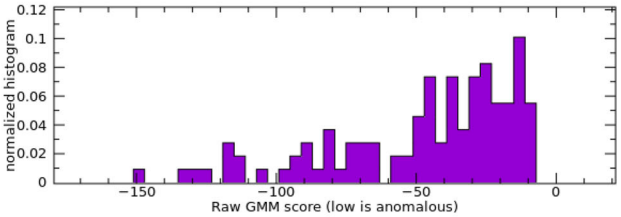
Analytic	GMM-KNN
Noise Correction Enabled	False
Report Format	GMM-KNN v1

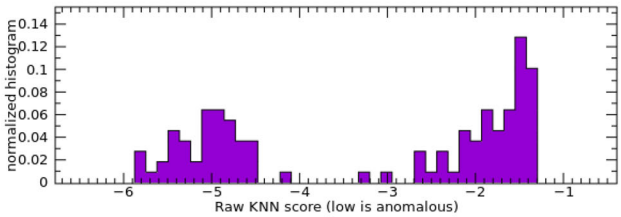
Summary

Number of images: 10

Image reports: EO1A0100612016271110KF_ALI_L1G EO1A0120292016271110P0_ALI_L1G EO1A0280342016271110P3_ALI_L1G EO1A0420352016271110KF_ALI_L1G EO1A0470122016271110KF_ALI_L1G EO1A0760132016271110K5_ALI_L1G EO1A0910842016271110KA_ALI_L1G EO1A0930632016271110KF_ALI_L1G EO1A1070342016271110K7_ALI_L1G EO1A1090822016271110KF_ALI_L1G

Most anomalous objects





Rank (GMM)	Rank (KNN)	Image	Object id	Raw GMM score	Raw KNN score	Clump size (pixels)
1	7	EO1A0930632016271110KF_ALI_L1G	_cluster_20	-150.525	-5.47704	10
2	20	EO1A0280342016271110P3_ALI_L1G	_cluster_26	-134.392	-5.08599	35
3	48	EO1A0280342016271110P3_ALI_L1G	_cluster_21	-127.8	-3.06031	31
4	12	EO1A0930632016271110KF_ALI_L1G	_cluster_19	-125.668	-5.34021	7
5	4	EO1A0910842016271110KA_ALI_L1G	_cluster_15	-118.148	-5.63969	10
6	2	EO1A0910842016271110KA_ALI_L1G	_cluster_25	-117.62	-5.80401	7

Fig. 3 An example GMM–KNN wheel analytic report summary from the end of September, 2016. This analytic has processed a batch of 10 new scenes from the EO-1 ALI instrument, identified and scored anomalous spectral objects as described in the text, and presented a rank ordering of the most interesting features from this wheel batch as a summary report

training data from two classes [24]. The result of this analytic is a GeoTiff showing the classification at each pixel.

4.6.1 Building a training dataset

We constructed a training dataset of classified spectra from sections of EO-1 Hyperion images over areas with known land coverage and cloud coverage and confirmed our selections by visual inspection of three-color (RGB) images created of the training images. We used a combination of Hyperion bands B16 (508.22 nm), B23 (579.45 nm), and B29 (640.5 nm) to construct the RGB images. For each image contributing to the training dataset, we only included spectra for collections of pixels that were visually confirmed as exclusively desert, water, clouds, or vegetation.

The training dataset consists of approximately 6,000 to 9,000 objects for each class. We include spectra for a variety of different regions on the Earth observed during different times of the year and a range of solar elevation angles. Because absolute values are necessary to directly compare the training data to all test images, we convert the raw irradiance values from the Level 1G data to at-sensor reflectance using Eq. 1. Table 2 lists general properties of the Hyperion scenes used in constructing the training set, where the class column indicates the class(es) (C = cloud, W = water, V = vegetation, and D = desert) to which that scene contributed.

In Fig. 4, we show a plot of the average reflectance spectra for each of the four classes in our training set. These spectra are in good agreement with the spectral signatures of cloud, vegetation, and desert sand presented in other examples of EO-1 Hyperion data analysis, specifically Figures 3 and 4 in Griffin et. al. [12,25].

4.6.2 Classifier results

We show a visual comparison of cloud coverage determined by our classifier with cloud coverage amounts stated by EarthExplorer for three Hyperion scenes of the big island of Hawaii with varying amounts of clouds and an additional scene of a section of the coast of Valencia. In Fig. 5, we show the resulting image report of the classifier analytic. This report shows a classification of an ALI observation of the Danube River in the Szekszard region of Hungary, with the river water in the classified overlay colored as blue, veg-

Table 2 Scenes included in training set

Region name	Class	Obs Date	Sun Azim. (°)	Sun Elev. (°)
Aira	W	4/18/14	119.12	49.1
San Rossore	C/W	1/29/14	145.5	20.9
San Rossore	C/V	8/10/12	135.8	54.5
Barton Bendish	C	8/22/13	142.7	43.5
Jasper Ridge	V/D	9/17/13	140.7	46.9
Jasper Ridge	V/C	9/14/13	132.9	45.2
Jasper Ridge	V/C	9/27/12	147.6	45.4
Arabian Desert	D	12/30/12	147.0	29.9
Jornada	D	12/10/12	28.7	151.4
Jornada	D	7/24/12	59.6	107.5
Negev	D	9/15/12	130.7	52.1
White Sands	C	7/29/12	58.3	108.4
Besetsutzuyu	V	7/14/12	56.8	135.5
Kenatedo	W	6/22/12	50.5	46.5
Santarem	W	6/17/12	57.1	118.15
Bibubemuku	W	5/20/12	57.5	127.7

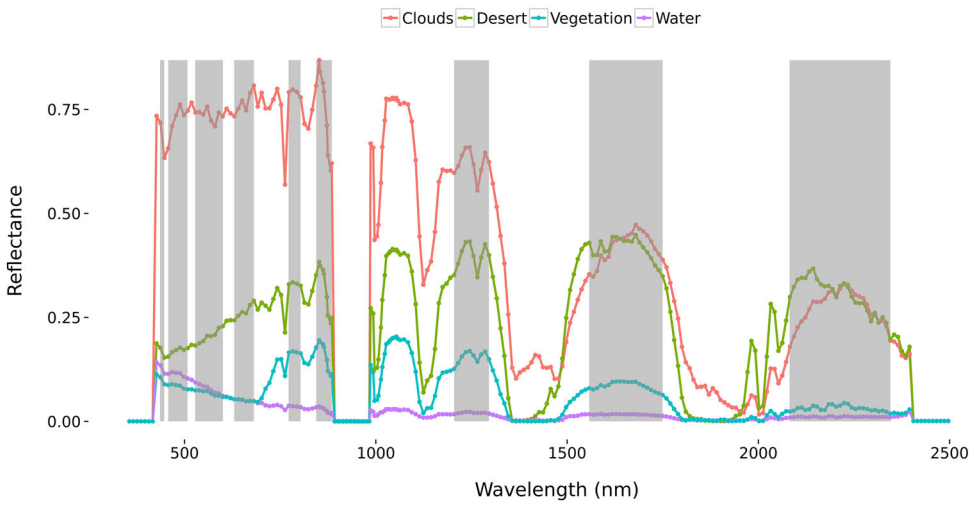


Fig. 4 The average reflectance spectra for each of the four classifications in the training data used by our implementation of the Supervised Spectral Classifier analytic. The four classes are clouds (*salmon*), desert (*lime green*), vegetation (*cyan*), and water (*purple*). Shaded *gray areas* show the wavelength coverage of ALI bands, which are the wavelength regions used by the classifier described

etation colored green, dry land colored brown, and cloud colored white. This classifier has some difficulty in distinguishing between clouds and ground water, as can be seen with the white pixels following the Danube River path.

To further confirm that the classifier analytic is generating reasonable results, we compare the fractional amount of land coverage types calculated by the classifier with known fractional amounts from other sources. We compare our results for classified cloud coverage with the cloud coverage amounts stated for individual scenes available through the EarthExplorer tool from the US Geological Survey (USGS) [26]. In Fig. 6, we show a plot comparing expected cloud and water coverage to the coverages determined by our classifier for 20 random test scenes. For each scene, the expected cloud coverage is taken as the center of the range provided by the USGS EarthExplorer summary for that image. The expected water coverage is calculated from scenes of islands that are completely contained within the image. We can then calculate expected water coverage by removing the known fractional land area of the islands and the USGS reported cloud coverage. We fit a regression line to the data, which shows an overall consistent relationship between the classified results and expected estimates.

4.7 Viewing analytic results

For convenience, each analytic produces a report after each run of the wheel. These reports are built from the JSON results stored in Accumulo and are accessible to the public via a web page. The generated reports contain spectral and geospatial information about the scene analyzed as well as analytic results. An overview summary report is created for

Matsu Analytic Image Report

Image	EO1A1880282016182110KF_ALI_L1G
Collection Date	2016-06-30 (day 182)
Analysis Date	Fri Jul 1 08:54:17 2016
Analytic Environment	
Analytic	SVMClassifier-v1
Summary Stats	ss-2013-12-r1
Data Ingest	populateHDFS-2013-11-r1
Report Format	reportSVMClassifierV1

Terrain Type	Area (Pixels)	Area (Percent)	Area (m^2)	Color
Cloud	10738	0.29%	9664200	COLOR
Desert	739304	19.77%	665373600	COLOR
Water	39752	1.06%	35776800	COLOR
Vegetation	2948896	78.88%	2654006400	COLOR

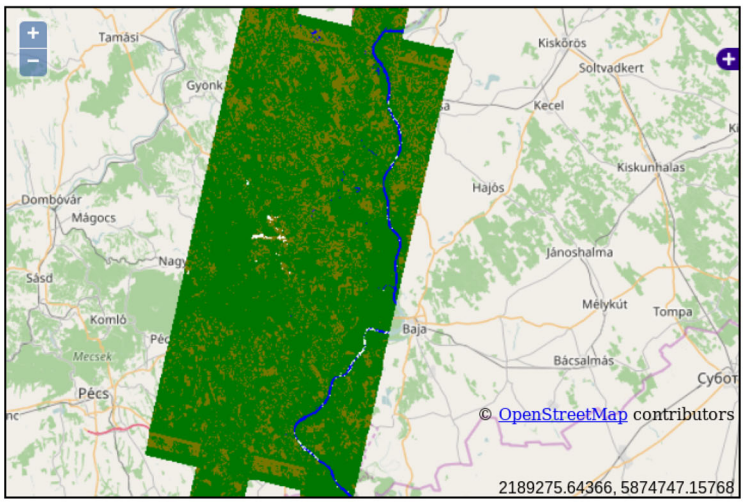


Fig. 5 An example image report for the Supervised Spectral Classifier

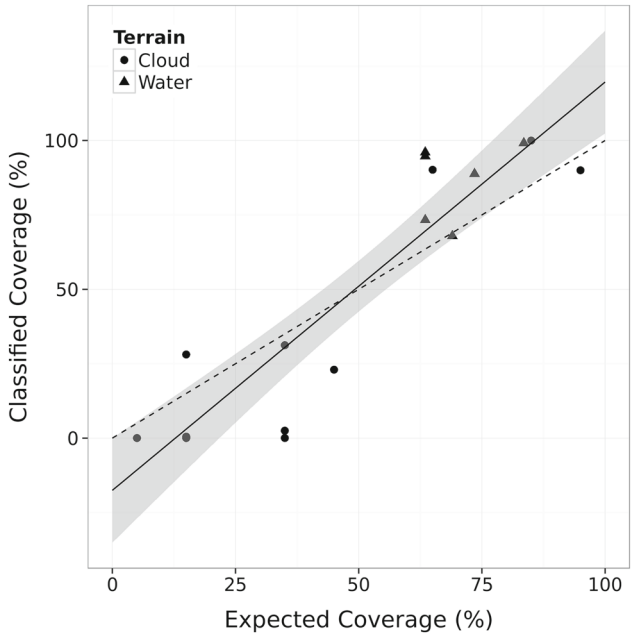


Fig. 6 Comparison of expected cloud and water coverages from USGS vs. coverages calculated from our classifier. Expected water points (*triangles*) are calculated from island scenes as described in the text. Expected cloud coverage estimates (*circles*) are taken from USGS EarthExplorer quoted cloud coverage for each image. The linear regression is the *solid black line*, and the *gray shaded area* is the 95% confidence interval. A 1–1 relationship is shown as a *dashed black line* for comparison

all daily data processed by an analytic in one run of the wheel in addition to reports for individual scenes. These reports are generated and viewable immediately upon completion of the scan of new data available each day at the following address: https://matsu-analytics.opensciencedatacloud.org. Analytic products are also made programmatically accessible through a Web Map Service.

5 Discussion

In our application of an analytic wheel, the preprocessing and bundling of common operations provides significant efficiency for the subsequent analytics over the Earth Observing-1 data. For example, the atmospheric and other corrections applied to Level 0 Hyperion data and the conversion to GeoTiff file format for all bands in Hyperion can take 30 min to over an hour for a single scene on a 32-core OSDC virtual machine. With even just a dozen scenes per day and five analytics in our current system, the wheel saves significant overhead in terms of time to process.

In terms of scaling up the number of analytics, as many analytics can be pushed into the wheel as can run in a given time frame, and data generation may outpace the wheel revolution. However, this problem is not unique to the wheel framework, but in any processing pipeline that may have some finite compute capacity. If certain analytics have more stringent time constraints than others, they can be set to run in the wheel earlier than others. There is also flexibility in the types of analytics that can be placed in the wheel. In the example analytics we have provided here, all take as input a batch of EO-1 scenes, though some act on individual scenes within the batch (like the classifier analytic) and others ana-

lyze the products of all scenes in a batch (for example, the rank ordering of all anomalous objects across scenes in a daily batch of EO-1 data).

In terms of analytics, the Matsu Wheel allows for additional analytics to be easily slotted in with no change to the existing framework so that we can continue to develop a variety of scanning analytics over these data. We are extending our existing Supervised Spectral Classifier to use specifically over floodplain regions to aid in flood detection for disaster relief. We are also planning to develop a similar analytic to aid in the detection of fires.

The analytics we described here are all detection algorithms, but we can also apply this framework and the results of our current analytics to implement algorithms for prediction. For example, our future work includes developing Wheel analytics for the prediction of floods. This could be done using the following approach:

1. Develop a dataset of features describing the observed topology of the Earth.
2. Use the topological data to identify "flood basins," or regions that may accumulate water around a local minimum.
3. Determine the relationship between detected water coverage in flood basins and the volume of water present.
4. Use observed water coverage on specific dates to relate the water volume in flood basins with time.
5. Use geospatial climate data to relate recent rainfall amounts with water volume, which then provides a simple model relating rainfall to expected water coverage at any pixel.

This proposed scanning analytic would provide important information particularly if implemented over satellite data with global and frequent coverage, such as data from the Global Precipitation Measurement (GPM) mission [27,28]. Our future work also involves continuing to develop the Matsu Wheel analytics and apply this framework to additional Earth satellite datasets.

6 Summary

We have described here the Project Matsu Wheel, which is what we believe to be the first working application of a Hadoop-based framework for creating analysis products from a daily application of scanning queries to satellite imagery data. This system is unique in that it allows for new analytics to be dropped into a daily process that scans all available data and produces new data analysis products. With an analytic wheel scanning framework, the data need to be read in only once, regardless of the number or types of analytics applied, which is particularly advantageous when large volumes of data, such as those produced by Earth satellite observations, need to be processed or reprocessed by an assortment of analytics.

We currently use the Matsu Wheel to process daily spectral data from NASA's EO-1 satellite and make the data and wheel analytic products available to the public through the Open Science Data Cloud and via analytic reports on the web.

A driving goal of Project Matsu is to develop open source technology for satellite imagery analysis and data mining analytics to provide data products in support of human-assisted disaster relief. The open nature of this project and its implementation over commodity hardware encourage the development and growth of a community of contributors to develop new scanning analytics that can be dropped into the analytic wheel for processing these and other Earth satellite data.

Acknowledgements Project Matsu is an Open Commons Consortium (OCC)-sponsored project supported by the Open Science Data Cloud. The source code and documentation are made available on GitHub at (https://github.com/LabAdvComp/matsu-project). This work was supported in part by grants from Gordon and Betty Moore Foundation and the National Science Foundation (Grant OISE 1129076 and CISE 1127316). The Earth Observing-1 satellite image is courtesy of the Earth Observing-1 project team at NASA Goddard Space Flight Center. The EarthExplorer cloud coverage calculations are available from the US Geological Survey on earthexplorer.usgs.gov.

References

1. Patterson, M.T., Anderson, N., Bennett, C., Bruggemann, J., Grossman, R.L., Handy, M., Ly, V., Mandl, D.J., Pederson, S., Pivarski, J., Powell, R., Spring, J., Wells, W., Xia, J.: In: 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), pp. 156–165 (2016). doi:10.1109/BigDataService.2016.39
2. Grossman, R.L., Heath, A., Murphy, M., Patterson, M., Wells, W.: A Case for Data Commons: Toward Data Science as a Service. Computing in Science & Engineering **18**(5), 10 (2016). doi:10.1109/MCSE.2016.92. http://scitation.aip.org/content/aip/journal/cise/18/5/10.1109/MCSE.2016.92
3. Hearn, D., Digenis, C., Lencioni, D., Mendenhall, J., Evans, J.B., Welsh, R.D.: In: Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International, vol. 2, pp. 897–900 (2001). doi:10.1109/IGARSS.2001.976673
4. Pearlman, J., Carman, S., Segal, C., Jarecke, P., Clancy, P., Browne, W.: In: Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International, vol. 7, pp. 3036–3038 (2001). doi:10.1109/IGARSS.2001.978246
5. Backman, N., Pattabiraman, K., Fonseca, R., Cetintemel, U.: In: Proceedings of Third International Workshop on MapReduce and Its Applications Date (ACM), pp. 1–8 (2012)

6. Zeng, J., Plale, B.: In: eScience (eScience), 2013 IEEE 9th International Conference on (IEEE), pp. 164–171 (2013)
7. Kienzler, R., Bruggmann, R., Ranganathan, A., Tatbul, N.: In: Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on (IEEE), pp. 159–166 (2012)
8. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (ACM, New York, NY, USA), SIGMOD '08, pp. 1099–1110 (2008). doi:10.1145/1376616.1376726.
9. Islam, M., Huang, A.K., Battisha, M., Chiang, M., Srinivasan, S., Peters, C., Neumann, A., Abdelnur, A.: In: Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies (ACM), p. 4 (2012)
10. http://spark.apache.org/
11. Jackson, K.R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H.J., Wright, N.J.: In: CloudCom, pp. 159–168 (2010)
12. Griffin, M.K., Hsu, S.M., Burke, H.H., Orloff, S.M., Upham, C.A.: Examples of EO-1 Hyperion data analysis. Linc. Lab. J. **15**(2), 271 (2005)
13. Goodenough, D.G., Dyk, A., Niemann, K.O., Pearlman, J.S., Chen, H., Han, T., Murdoch, M., West, C.: Processing Hyperion and ALI for forest classification. IEEE Trans. Geosci. Remote Sens. **41**(6), 1321 (2003)
14. Griggin, M., Burke, H., Mandl, D., Miller, J.: In: IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477), vol. 1, pp. 86–89 (2003). doi:10.1109/IGARSS.2003.1293687
15. Wang, K., Franklin, S.E., Guo, X., Cattet, M.: Remote sensing of ecology, biodiversity and conservation: a review from the perspective of remote sensing specialists. Sensors **10**(11), 9647 (2010)
16. Matteoli, S., Diani, M., Corsini, G.: A tutorial overview of anomaly detection in hyperspectral images. IEEE Aerosp. Electron. Syst. Mag. **25**(7), 5 (2010)
17. Grossman, R.L., Greenway, M., Heath, A.P., Powell, R., Suarez, R.D., Wells, W., White, K.P., Atkinson, M.P., Klampanos, I.A., Alvarez, H.L., Harvey, C., Mambretti, J.: In: SC Companion, pp. 1051–1057 (2012)
18. White, T.: Hadoop—The Definitive Guide: Storage and Analysis at Internet Scale, 3rd edn, revised and updated. O'Reilly (2012)
19. Dean, J., Ghemawat, S.: In: OSDI, pp. 137–150 (2004)
20. https://accumulo.apache.org/
21. Borovica, R., Idreos, S., Ailamaki, A., Zukowski, M., Fraser, C.: Smooth Scan: One Access Path to Rule them all. Accessed from http://stratos.seas.harvard.edu/files/stratos/files/smoothscan.pdf (2017)
22. Ip, F., Dohm, J., Baker, V., Doggett, T., Davies, A., Castao, R., Chien, S., Cichy, B., Greeley, R., Sherwood, R., Tran, D., Rabideau, G.: Flood detection and monitoring with the Autonomous Sciencecraft Experiment onboard EO-1. Remote Sensing of Environment **101**(4), 463 (2006). doi:10.1016/j.rse.2005.12.018. http://www.sciencedirect.com/science/article/pii/S0034425706000228
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825 (2011)
24. Knerr, S., Personnaz, L., Dreyfus, G.: In: Neurocomputing. Springer, pp. 41–50 (1990)
25. Hua, H., Burke, K., Hsu, S., Griffin, M.K., Upham, C.A., Farrar, K.: In: IGARSS, pp. 1483–1486 (2004)
26. http://earthexplorer.usgs.gov/
27. http://www.nasa.gov/mission_pages/GPM/main/index.html
28. Neeck, S.P., Kakar, R.K., Azarbarzin, A.A., Hou, A.Y.: In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 8889 (2013). doi:10.1117/12.2031431